LISTING OF CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Previously presented) A method comprising:

detecting that a first virtual machine is attempting to transmit data to a second virtual machine;

mapping a transmitting memory element of the first virtual machine to a shared physical memory element;

mapping a receiving memory element of the second virtual machine to the shared physical memory element;

placing data from the first virtual machine into the shared physical memory element via the transmitting memory element; and

receiving the placed data from the shared physical memory element into the second virtual machine via the receiving memory element.

2. (Original) The method of claim 1, further including:

detecting that the first virtual machine has placed data in the shared physical memory element; and

informing the second virtual machine that data is available in the shared physical memory element.

3. (Original) The method of claim 1, further comprising:

detecting if the first virtual machine is attempting to transmit data to a non-virtual machine;

dynamically remapping the transmitting memory element of the first virtual

machine to a physical device associated with the transmitting memory element.

4. (Previously presented) The method of claim 3, wherein the transmitting memory

element of the first virtual machine is part of a first virtual device; and

the receiving memory element of the second virtual machine is part of a second

virtual device.

5. (Original) The method of claim 4, wherein first virtual device and the second virtual

device are devices selected from a group including:

an Ethernet device,

a network interface,

an audio device,

a storage device, and

a video device.

6. (Original) The method of claim 4, wherein the shared physical memory element is a

direct memory access (DMA) buffer.

7. (Previously presented) The method of claim 1, wherein detecting that a first virtual

machine is attempting to transmit data to a second virtual machine includes:

monitoring the first virtual machine;

comparing a destination of any data transmitted by the first virtual machine to an

address associated with the second virtual machine.

8. (Previously presented) The method of claim 1, wherein detecting that a first virtual machine is attempting to transmit data to a second virtual machine includes:

reading a mapping configuration data that specifies default virtual device to physical device mappings;

comparing the mapping configuration data for the first virtual machine to the mapping configuration data of the second virtual machine;

assuming that the first virtual machine is attempting to transmit data to the second virtual machine, if a transmitting virtual device of the first virtual machine is mapped to the same physical device as the receiving virtual device of the second virtual machine.

9. (Original) The method of claim 7, wherein mapping a transmitting memory element of the first virtual machine to a shared physical memory element includes:

determining if the transmitting memory element is currently mapped to a transmitting memory element of a physical device;

if so, unmapping of the transmitting memory element from the transmitting memory element of the physical device; and

remapping the transmitting memory element of the first virtual machine to the shared physical memory element.

10. (Original) The method of claim 9, further including:

detecting that the second virtual machine is attempting to transmit data to the first virtual machine;

mapping a transmitting memory element of the second virtual machine to the shared physical memory element; and

mapping a receiving memory element of the first virtual machine to the shared

physical memory element.

11. (Original) An article comprising:

a machine accessible medium having a plurality of machine accessible instructions,

wherein when the instructions are executed, the instructions provide for:

    detecting that a first virtual machine is attempting to transmit data to a second

virtual machine;

    mapping a transmitting memory element of the first virtual machine to a shared

physical memory element; and

    mapping a receiving memory element of the second virtual machine to the shared

physical memory element.

12. (Original) The article of claim 11, further including instructions providing for:

    detecting that the first virtual machine has placed data in the shared physical

memory element; and

    informing the second virtual machine that data is available in the shared physical

memory element.

13. (Original) The article of claim 11, further comprising instructions providing for:

    detecting if the first virtual machine is attempting to transmit data to a non-virtual

machine:

    dynamically remapping the transmitting memory element of the first virtual

machine to a physical device associated with the transmitting memory element.

14. (Original) The article of claim 13, wherein the transmitting buffer of the first virtual

machine is part of a first virtual device; and

the receiving buffer of the second virtual machine is part of a second virtual

device.

15. (Original) The article of claim 14, wherein first virtual device and the second virtual

device are devices selected from a group including:

an Ethernet device,

a network interface,

an audio device,

a storage device, and

a video device.

16. (Original) The article of claim 14, wherein the shared physical memory element is a

direct memory access (DMA) buffer.

17. (Previously presented) The article of claim 11, wherein the instructions provide for

detecting that a first virtual machine is attempting to transmit data to a second virtual

machine includes instructions providing for:

monitoring the first virtual machine;

comparing a destination of any data transmitted by the first virtual machine to an

address associated with the second virtual machine.

18. (Original) The article of claim 11, wherein the instructions provide for detecting that

a first virtual machine is attempting to transmit data to a second virtual machine includes

instructions providing for:

reading a mapping configuration data that specifies default virtual device to physical device mappings;

comparing the mapping configuration data for the first virtual machine to the mapping configuration data of the second virtual machine;

assuming that the first virtual machine is attempting to transmit data to the second virtual machine, if a transmitting virtual device of the first machine is mapped to the same physical device as the receiving virtual device of the second virtual machine.

19. (Original) The article of claim 17, wherein the instructions provide for mapping a transmitting memory element of the first virtual machine to a shared physical memory element includes instructions providing for:

determining if the transmitting memory element is currently mapped to a transmitting memory element of a physical device;

if so, unmapping of the transmitting memory element from the transmitting memory element of the physical device; and

remapping the transmitting memory element of the first virtual machine to the shared physical memory element.

20. (Original) The article of claim 19, further including instructions providing for:

detecting that the second virtual machine is attempting to transmit data to the first virtual machine;

mapping a transmitting memory element of the second virtual machine to the shared physical memory element; and

mapping a receiving memory element of the first virtual machine to the shared

physical memory element.

21-29. (Canceled)

30. (Canceled)

31. (Original) A system comprising:

a first virtual machine, having a first virtual device that includes a first virtual memory

element;

a second virtual machine, having a second virtual device that includes a second virtual

memory element;

a shared physical memory element; and

a virtual machine manager, having

        a cross-talk detector to detect if a first virtual machine is attempting to transmit

data to a second virtual machine; and

        a dynamic memory remapper to, if instructed by the cross-talk detector, map a

first virtual memory element of the first virtual machine to a second virtual memory

element of the second virtual machine via a shared physical memory element.

32. (Original) The system of claim 31, wherein the dynamic memory remapper is

capable of:

        mapping a transmitting memory element of the first virtual machine to a shared

physical memory element; and

mapping a receiving memory element of the second virtual machine to the shared

physical memory element.

33. (Original) The system of claim 32, wherein the cross-talk detector is further capable

of:

detecting that the first virtual machine has placed data in the shared physical

memory element; and

informing the second virtual machine that data is available in the shared physical

memory element.

34. (Original) The system of claim 32, wherein the cross-talk detector is further capable

of detecting if the first virtual machine is attempting to transmit data to a non-virtual

machine; and

the dynamic memory remapper is further capable of dynamically remapping the

transmitting memory element of the first virtual machine to a physical device associated

with the transmitting memory element.

35. (Original) The system of claim 34, wherein the cross-talk detector is further capable

of monitoring the first and second memories wherein the memories are part of virtual

devices selected from a group including:

an Ethernet device,

a network interface,

an audio device,

a storage device, and

a video device.

36. (Original) The system of claim 35, wherein the shared physical memory element is a direct memory access (DMA) buffer.

37. (Previously presented) The system of claim 36, wherein the cross-talk detector is capable of:

monitoring the first virtual machine;

comparing a destination of any data transmitted by the first virtual machine to an address associated with the second virtual machine.

38. (Original) The system of claim 32, wherein the dynamic memory remapper is capable of:

determining if the transmitting memory element is currently mapped to a transmitting memory element of a physical device;

if so, unmapping of the transmitting memory element from the transmitting memory element of the physical device; and

remapping the transmitting memory element of the first virtual machine to the shared physical memory element.

39. (Original) The system of claim 38, wherein the cross-talk detector is capable of detecting that the second virtual machine is attempting to transmit data to the first virtual machine; and

wherein the dynamic memory remapper is capable of mapping a transmitting memory element of the second virtual machine to the shared physical memory element; and

mapping a receiving memory element of the first virtual machine to the shared physical memory element.

40. (Original) The system of claim 31, wherein the first virtual memory element of the first virtual machine to a second virtual memory element of the second virtual machine are not identical but share substantially similar characteristics.

41. (Previously presented) A method of communicating between two virtual machines utilizing a virtual machine manager comprising:

    detecting that a first virtual machine, having a first virtual network interface, is attempting to transmit data to a second virtual machine, having a second virtual network interface, via the virtual network interfaces;

    mapping a transmitting memory element of the first virtual network interface to a first direct memory access buffer;

    mapping a receiving memory element of the second virtual network interface to the first direct memory access buffer;

    placing data from the first virtual machine into the first direct memory access buffer via the transmitting memory element of the first virtual network interface;
    receiving the placed data from the first direct memory access buffer into the second virtual machine via the receiving memory element of the second virtual network interface;

    mapping a receiving memory element of the first virtual network interface to a second direct memory access buffer;

    mapping a transmitting memory element of the second virtual network interface to the second direct memory access buffer;

    placing data from the second virtual machine into the second direct memory access buffer via the transmitting memory element of the second virtual network interface; and

receiving the placed data from the second direct memory access buffer into the first virtual machine via the receiving memory element of the first virtual network interface.

42. (Previously presented) The method of claim 41, further including:

detecting that the first virtual machine has placed data in the first direct memory access buffer;

informing the second virtual machine that data is available in the first direct memory access buffer;

detecting that the second virtual machine has placed data in the second direct memory access buffer; and

informing the first virtual machine that data is available in the second direct memory access buffer.

43. (Previously presented) The method of claim 41, further comprising the first virtual machine:

placing at least one packet into the first direct memory access buffer; and

moving a tail register of the first virtual network interface to indicate how many packets where written to the first direct memory access buffer.

44. (Previously presented) The method of claim 43, further comprising the virtual machine manager:

moving a receive descriptor head register of the second virtual network interface by a number of packets written to the first direct memory access buffer;

updating the status of the second network interface to indicate that a packet has been received; and

sending a receive interrupt to the second virtual machine.

45. (Previously presented) The method of claim 44, further comprising the second virtual machine:

      reading a packet from the first direct memory access buffer.

46. (Previously presented) The method of claim 45, further comprising, the virtual machine manager:

      detecting that the second virtual machine has read the packet from the first direct memory buffer;

      updating the status of the first virtual network interface to indicate that the packet has been received; and

      injecting a transmit complete interrupt to the first virtual machine.

47. (Presently presented) The method of claim 41, wherein detecting that a first virtual machine is attempting to transmit data to a second virtual machine includes:

      monitoring the first virtual machine;

      comparing a destination of any data transmitted by the first virtual machine to an address associated with the second virtual machine.

48. (Previously presented) A method comprising:

      detecting that a first virtual machine is configured to transmit data to a second virtual machine;

      statically mapping a transmitting memory element of the first virtual machine to a shared physical memory element;

      statically mapping a receiving memory element of the second virtual machine to the shared physical memory element;

placing data from the first virtual machine into the shared physical memory

element via the transmitting memory element; and

receiving the placed data from the shared physical memory element into the

second virtual machine via the receiving memory element.

49. (Previously presented) The method of claim 48, further comprising:

statically mapping a receiving memory element of the first virtual machine to a

second shared physical memory element;

statically mapping a transmitting memory element of the second virtual machine

to the second shared physical memory element;

placing data from the second virtual machine into the second shared physical

memory element via the transmitting memory element of the second virtual machine; and

receiving the placed data from the shared physical memory element into the first virtual

machine via the receiving memory element of the first virtual machine.

50. (Previously presented) The method of claim 48, wherein detecting that a first virtual

machine is configured to transmit data to a second virtual machine is performed when the

first virtual machine is started.

51. (Original) The method of claim 50, wherein detecting that a first virtual machine is

configured to transmit data to a second virtual machine includes reading a configuration

file that explicitly denotes that the first and second virtual machines are virtually coupled.

52. (Original) The method of claim 50, wherein detecting that a first virtual machine is configured to transmit data to a second virtual machine includes reading a configuration file that implicitly denotes that the first and second virtual machines are virtually coupled.

53. (Original) The method of claim 50, wherein the shared physical memory element comprises a direct access memory buffer.

54. (Original) The method of claim 53, wherein the virtual memory elements of the first and second virtual machines are part of virtual devices selected from a group of virtual devices comprising:

      an Ethernet device;

      a network device;

      an audio device; a storage device; and

      a video device.

55. (Original) An article comprising:

a machine accessible medium having a plurality of machine accessible instructions, wherein when the instructions are executed, the instructions provide for:

      detecting that a first virtual machine is configured to transmit data to a second virtual machine;

      statically mapping a transmitting memory element of the first virtual machine to a shared physical memory element; and

statically mapping a receiving memory element of the second virtual machine to

the shared physical memory element.

56. (Original) The article of claim 55, further comprising instructions providing for:

statically mapping a receiving memory element of the first virtual machine to a

second shared physical memory element; and

statically mapping a transmitting memory element of the second virtual machine

to the second shared physical memory element.

57. (Original) The article of claim 55, wherein the instructions providing for detecting

that a first virtual machine is configured to transmit data to a second virtual machine are

executed when the first virtual machine is started.

58. (Original) The article of claim 57, wherein the instructions providing for detecting

that a first virtual machine is configured to transmit data to a second virtual machine

includes instructions providing for reading a configuration file that explicitly denotes that

the first and second virtual machines are virtually coupled.

59. (Original) The article of claim 57, wherein the instructions providing for detecting

that a first virtual machine is configured to transmit data to a second virtual machine

includes instructions providing for reading a configuration file that implicitly denotes that

the first and second virtual machines are virtually coupled.

60. (Original) The article of claim 57, wherein the instructions provide for the shared

physical memory element comprising a direct access memory buffer.

61. (Original) The article of claim 60, wherein the virtual memory elements of the first

and second virtual machines are part of virtual devices selected from a group of virtual

devices comprising:

an Ethernet device;

a network device;

an audio device; a storage device; and

a video device.